

Uvod v *Mathematico*

Nekaj starega, nekaj novega

Osnovna uporaba

Vgrajene funkcije

Simbolico racunanje

Seznami

```
In[ ]:= seznam = {1, 2, 3, "FMF", x^10}
```

```
Out[ ]:=  
      {1, 2, 3, FMF, x10}
```

```
In[ ]:= seznam[[3]]
```

```
Out[ ]:=  
      3
```

```
In[ ]:= seznam[[-2]]
```

```
Out[ ]:=  
      FMF
```

```
In[ ]:= seznam[[2 ;; 4]]
```

```
Out[ ]:=  
      {2, 3, FMF}
```

```
In[ ]:= Table[funkcija[x], {x, 3, 7}]
```

```
Out[ ]:=  
      {funkcija[3], funkcija[4], funkcija[5], funkcija[6], funkcija[7]}
```

Stevila

Mathematica racuna **eksaktno**.

```
In[ ]:= 1 / 7 + 2 / 3
```

```
Out[ ]:=  
      17  
      —  
      21
```

```
In[ ]:= N[%]
Out[ ]=
0.809524
```

Racuna tudi na poljubno natancnost.

```
In[ ]:= N[ $\pi$ , 500]
Out[ ]=
3.14159265358979323846264338327950288419716939937510582097494459230781640628620899862\
803482534211706798214808651328230664709384460955058223172535940812848111745028410270\
193852110555964462294895493038196442881097566593344612847564823378678316527120190914\
564856692346034861045432664821339360726024914127372458700660631558817488152092096282\
925409171536436789259036001133053054882046652138414695194151160943305727036575959195\
30921861173819326117931051185480744623799627495673518857527248912279381830119491
```

Nizi

Sklapljanje nizov:

```
In[ ]:= "To je niz" + "To je tudi niz"
Out[ ]=
To je niz + To je tudi niz
```

```
In[ ]:= "To je niz" <> "To je tudi niz"
Out[ ]=
To je nizTo je tudi niz
```

Vstavljanje v niz:

```
In[ ]:= StringReplace["Tole je niz.", "e" → "[E]"]
Out[ ]=
To[E] j[E] niz.
```

```
In[ ]:= StringForm["Racun je koncan, vrednosti rezultata so: a = ``, b = ``. Adijo!", 5,  $\pi$ ]
Out[ ]=
Racun je koncan, vrednosti rezultata so: a = 5, b =  $\pi$ . Adijo!
```

```
In[ ]:= TemplateApply["Vsota je enaka <* 1+1 *>."]
Out[ ]=
Vsota je enaka 2.
```

Spremenljivke in funkcije

Definicije

Mathematica razlikuje med = in :=.

```
In[ ]:= Now
Out[ ]=
```

Thu 25 Feb 2021 15:22:43 GMT+1.

```
In[*]:= mojaSpremenljivka = 7
casObIzracunu = Now
trenutniCas := Now
```

```
Out[*]=
```

```
7
```

```
Out[*]=
```

```
Thu 25 Feb 2021 15:22:51 GMT+1.
```

```
In[*]:= casObIzracunu
```

```
Out[*]=
```

```
Thu 25 Feb 2021 15:22:51 GMT+1.
```

```
In[*]:= trenutniCas
```

```
Out[*]=
```

```
Thu 25 Feb 2021 15:23:14 GMT+1.
```

Vrednosti spremenljivk pobrisemo z `=.`.

```
In[*]:= mojaSpremenljivka =.
```

```
In[*]:= Clear[mojaSpremenljivka]
```

Funkcije definiramo na naslednji način:

```
In[*]:= mojaFunkcija[x_, y_] := x + 2 y
```

```
In[*]:= mojaFunkcija[1, 1]
```

```
Out[*]=
```

```
3
```

```
In[*]:= mojaFunkcija[1, y]
```

```
Out[*]=
```

```
1 + 2 y
```

```
In[*]:= Clear[mojaFunkcija]
```

Prepisovalna pravila

Prepisovalno pravilo definiramo s puščico "`->`", naštejemo jih v zavutih oklepajih

```
In[*]:= x * y + 2 /. {x -> 2}
x * y + 2 /. {x -> 2, y -> b}
Sin[x] /. Sin -> Cos
```

```
Out[*]=
```

```
2 + 2 y
```

```
Out[*]=
```

```
2 + 2 b
```

```
Out[*]=
```

```
Cos[x]
```

```
In[*]:= 1 + x^2 + x^4 /. x^p_ => f[p]
```

```
Out[*]=
1 + f[2] + f[4]
```

Pravilo z vzorcem

```
In[*]:= 1 + x^2 + x^4 /. x^p_ -> f[p]
```

```
Out[*]=
1 + f[2] + f[4]
```

Več neodvisnih pravil

```
In[*]:= x /. {{x -> 1}, {x -> 3}, {x -> 7}}
```

```
Out[*]=
{1, 3, 7}
```

Kako bi napisali pravilo za odvod monoma?

```
In[*]:= prep = {x^y_ -> y x^(y - 1), x -> 1}
```

```
Out[*]=
{x^y_ -> x^(y-1) y, x -> 1}
```

```
In[*]:= x^5 /. prep (* enkratna uporaba prepisovalnega pravila *)
```

```
x^5 + 3 x^2 /. prep
```

```
x^5 //. prep (* večkratna uporaba prepisovalnega pravila *)
```

```
Out[*]=
5 x^4
```

```
Out[*]=
6 x + 5 x^4
```

```
Out[*]=
120
```

Anonimne funkcije

Lahko racunamo tudi s funkcijami, ki jim ne damo imena / jih ne shranimo v spremenljivko. Imenujemo jih **anonimne funkcije**. Parameter take funkcije oznacimo z # (ali z **#1**, **#2** ..., ce je vec parametrov), zakljucek funkcije pa z **&**.

```
In[*]:= 5 // # + 1 &
```

```
Out[*]=
6
```

```
In[*]:= # + 1 & [5]
```

```
Out[*]=
6
```

```
In[*]:= Clear[x]
```

```
In[*]:= #1 - #2 &[x, y]
```

```
Out[*]=
x - y
```

```
In[*]:= Select[Range[100], Mod[#, 7] == 0 &]
Out[*]=
{7, 14, 21, 28, 35, 42, 49, 56, 63, 70, 77, 84, 91, 98}
```

Aplikacije/uporabe funkcij

Funkcijo lahko uporabimo na vseh elementih seznama. Pri tem moramo biti pozorni na različne načine 'aplikacije'.

```
In[*]:= f[{x, y, z}]
          f@{x, y, z}
          Apply[f, {x, y, z}]
          f@@{x, y, z}
          Map[f, {x, y, z}]
          f /@ {x, y, z}

Out[*]=
f[{x, y, z}]

Out[*]=
f[{x, y, z}]

Out[*]=
f[x, y, z]

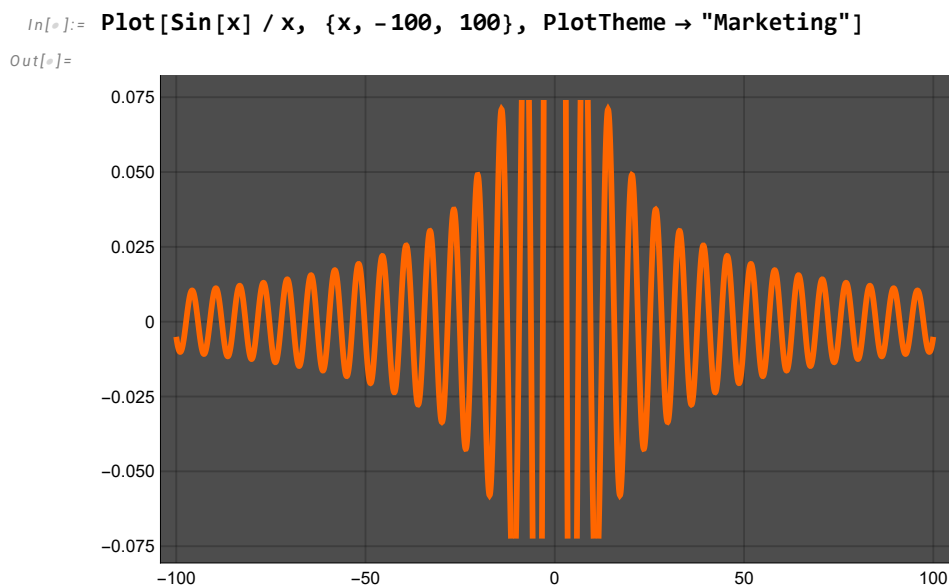
Out[*]=
f[x, y, z]

Out[*]=
{f[x], f[y], f[z]}

Out[*]=
{f[x], f[y], f[z]}
```

Opcijski parametri

Mnogo vgrajenih funkcij sprejme dodatne opsijske parametre, ki jih nastavimo z ->.



```
In[*]:= Options[Plot]
```

```
Out[*]=
```

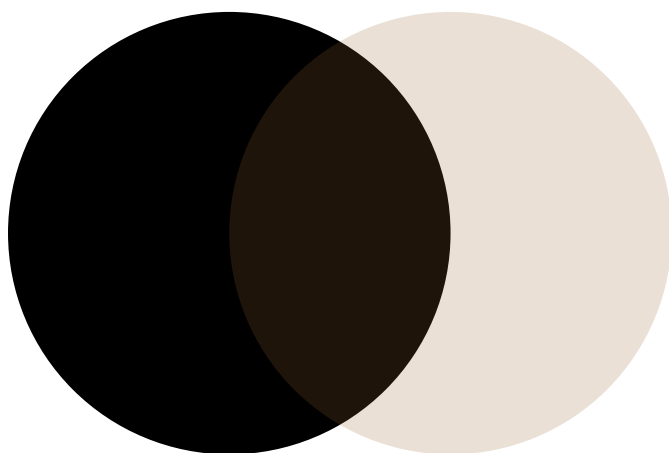
```
{AlignmentPoint → Center, AspectRatio →  $\frac{1}{\text{GoldenRatio}}$ , Axes → True, AxesLabel → None,
AxesOrigin → Automatic, AxesStyle → {}, Background → None, BaselinePosition → Automatic,
BaseStyle → {}, ClippingStyle → None, ColorFunction → Automatic,
ColorFunctionScaling → True, ColorOutput → Automatic, ContentSelectable → Automatic,
CoordinatesToolOptions → Automatic, DisplayFunction → $DisplayFunction, Epilog → {},
Evaluated → Automatic, EvaluationMonitor → None, Exclusions → Automatic,
ExclusionsStyle → None, Filling → None, FillingStyle → Automatic,
FormatType → TraditionalForm, Frame → False, FrameLabel → None, FrameStyle → {},
FrameTicks → Automatic, FrameTicksStyle → {}, GridLines → None,
GridLinesStyle → {}, ImageMargins → 0., ImagePadding → All, ImageSize → Automatic,
ImageSizeRaw → Automatic, LabelingSize → Automatic, LabelStyle → {},
MaxRecursion → Automatic, Mesh → None, MeshFunctions → {#1 &}, MeshShading → None,
MeshStyle → Automatic, Method → Automatic, PerformanceGoal → $PerformanceGoal,
PlotLabel → None, PlotLabels → None, PlotLegends → None, PlotPoints → Automatic,
PlotRange → {Full, Automatic}, PlotRangeClipping → True, PlotRangePadding → Automatic,
PlotRegion → Automatic, PlotStyle → Automatic, PlotTheme → $PlotTheme,
PreserveImageOptions → Automatic, Prolog → {}, RegionFunction → (True &),
RotateLabel → True, ScalingFunctions → None, TargetUnits → Automatic,
Ticks → Automatic, TicksStyle → {}, WorkingPrecision → MachinePrecision}
```

Grafika

Narisemo lahko karkoli.

```
In[*]:= Graphics[{Black, Disk[{0, 0}], Opacity[0.2], Brown, Disk[{1, 0}]}]
```

```
Out[*]=
```



Graficne elemente lahko izvozimo kot slike v razlicnih formatih.

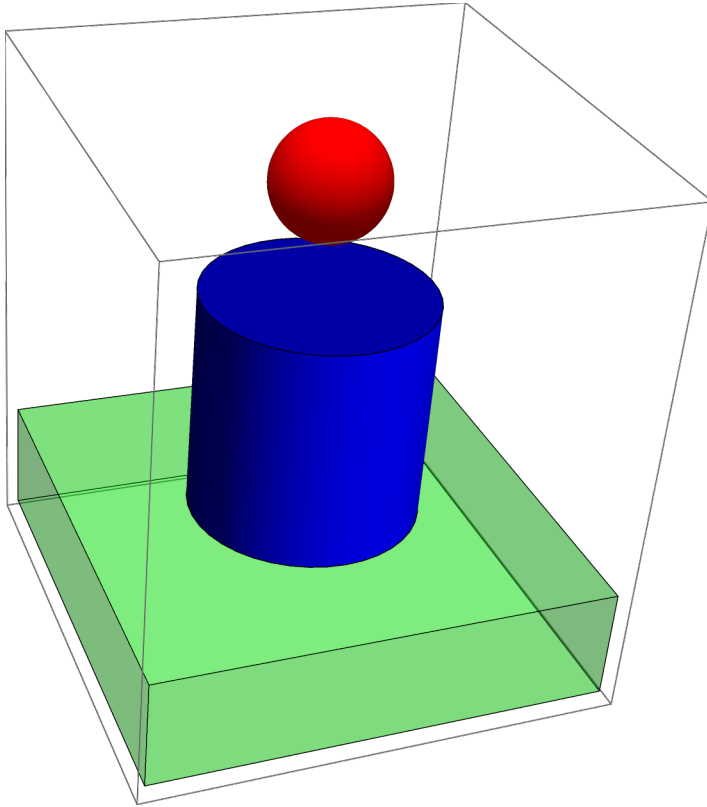
```
In[*]:= Export["~/Dropbox/pedagog/2020-orodja/slika.jpg", %, ImageResolution → 600]
```

```
Out[*]=
```

```
~/Dropbox/pedagog/2020-orodja/slika.jpg
```

Risemo lahko tudi 3D grafiko.

```
In[*]:= Graphics3D[{Blue, Cylinder[], Red, Sphere[{0, 0, 2}, 0.5],
  Green, Opacity[0.3], Cuboid[{-2, -2, -2}, {2, 2, -1}]}]
Out[*]=
```



Interaktivnost

Izraze lahko **manipuliramo**.

```
In[*]:= Plot[Sin[x], {x, -5, 5}]
In[*]:= Manipulate[Plot[Sin[a x], {x, -5, 5}], {a, 1, 10}]
```

Uvedemo lahko **dinamicne prikaze**.

```
In[*]:= x = 0
  y = 0
In[*]:= Dynamic[x]
In[*]:= Slider[Dynamic[x]]
In[*]:= Slider2D[{Dynamic[x], Dynamic[y]}, {-3, 3}]
In[*]:= Dynamic[Graphics[{Black, Disk[{0, 0}], Opacity[0.2], Brown, Disk[{x, y}]},
  PlotRange -> {{-3, 3}, {-3, 3}}]]
In[*]:= x = .
  y = .
```

If, for, while

Mathematica vse to sicer zna, ampak zna tudi toliko drugih stvari, da najbrz pocnete nekaj narobe, ce vasa koda vsebuje veliko **if, for, while ...**

```
In[*]:= If[x > 1, "Vecji", "Manjsi"] /. x -> 2
```

```
In[*]:= For[i = 0, i < 5, i++, Print[i]]
```

Kadar zelimo definirati funkcijo, pri kateri uporabljamo tudi lokalne spremenljivke, lahko uporabimo **Module**.

```
In[*]:= VsiKvadrati[n_] := Module[{seznam = {}, i = 0},
  While[i < n,
    i++;
    AppendTo[seznam, i^2];
  ];
  seznam
]
```

```
In[*]:= VsiKvadrati[10]
```

```
In[*]:= VsiKvadratiBolje[n_] := Table[i^2, {i, 1, n}]
```

```
In[*]:= VsiKvadratiBolje[10]
```

```
In[*]:= VsiKvadrati[10^4]; // Timing
```

```
In[*]:= VsiKvadratiBolje[10^4]; // Timing
```

Slovarji

Vnos z naravnim jezikom

Manipuliranje in analiza slik

Oblak